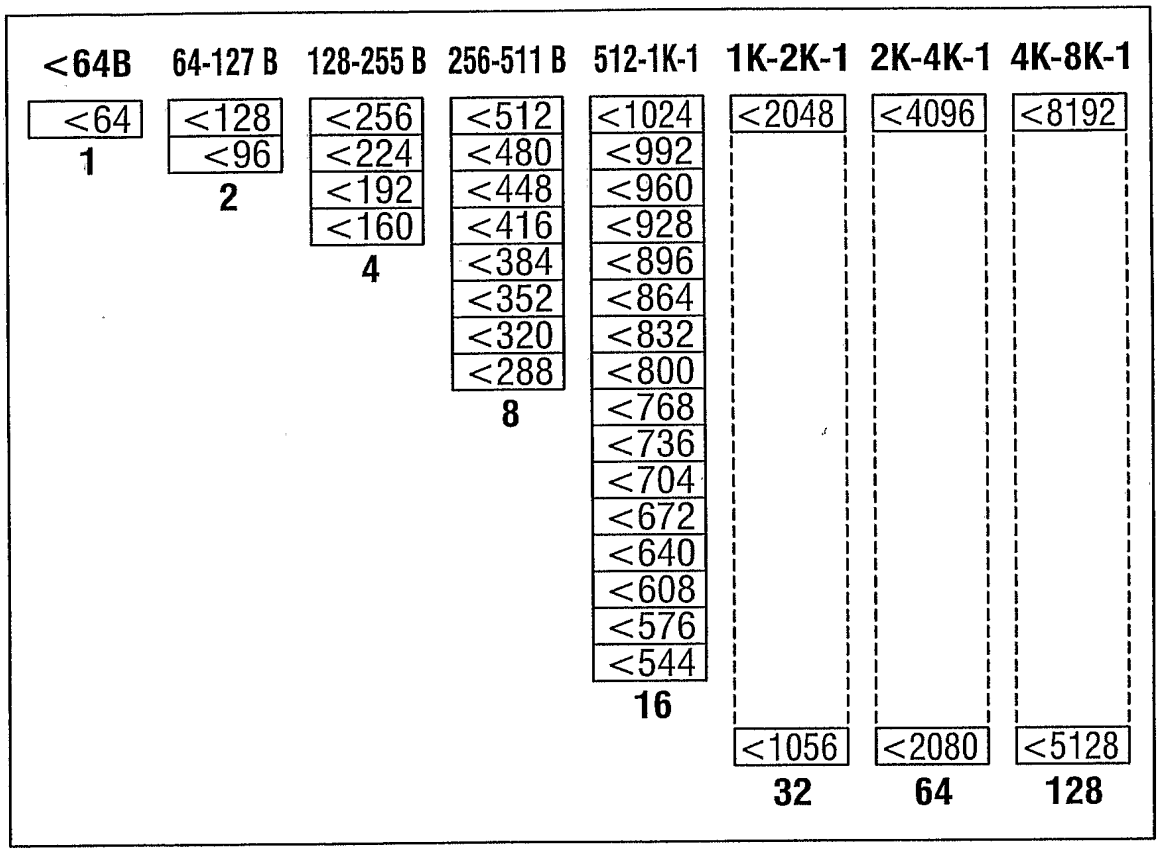
<64B
64-127 B
128-255 B
256-511 B
512-1K-1
1K-2K-1
2K-4K-1
4K-8K-1
<64
1
<128
<96
2
<256
<224
<192
<160
4
<512
<480
<448
<416
<384
<352
<320
<288
8
<1024
<992
<960
<928
<896
<864
<832
<800
<768
<736
<704
<672
<640
<608
<576
<544
16
<2048
<1056
32
<4096
<2080
64
<8192
<5128
128

Fig: 1

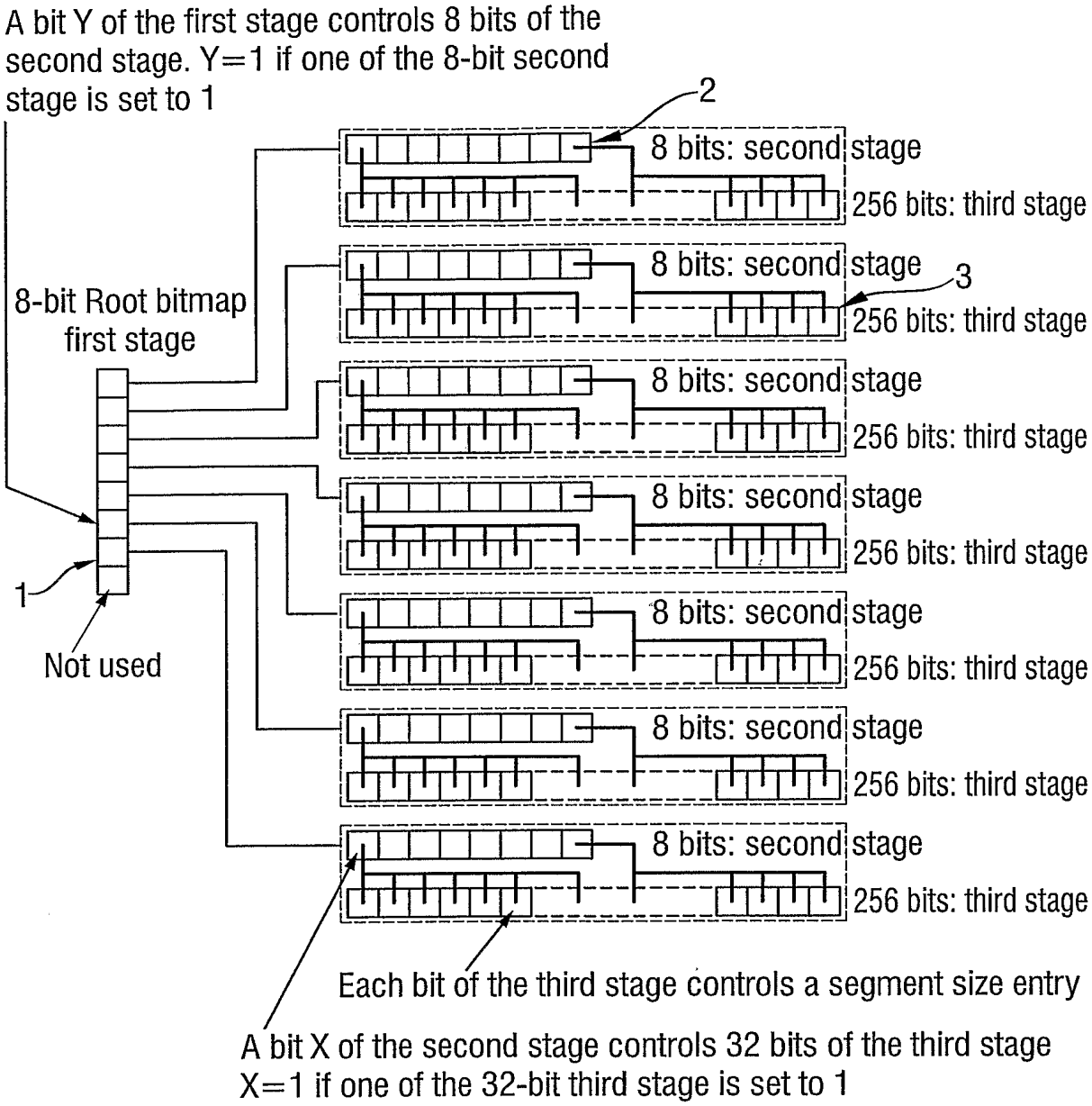
A bit Y of the first stage controls 8 bits of the second stage. Y=1 if one of the 8-bit second stage is set to 1
2
8 bits: second stage
256 bits: third stage
3
8-bit Root bitmap first stage
1
Not used
Each bit of the third stage controls a segment size entry
A bit X of the second stage controls 32 bits of the third stage
X=1 if one of the 32-bit third stage is set to 1

Fig: 2

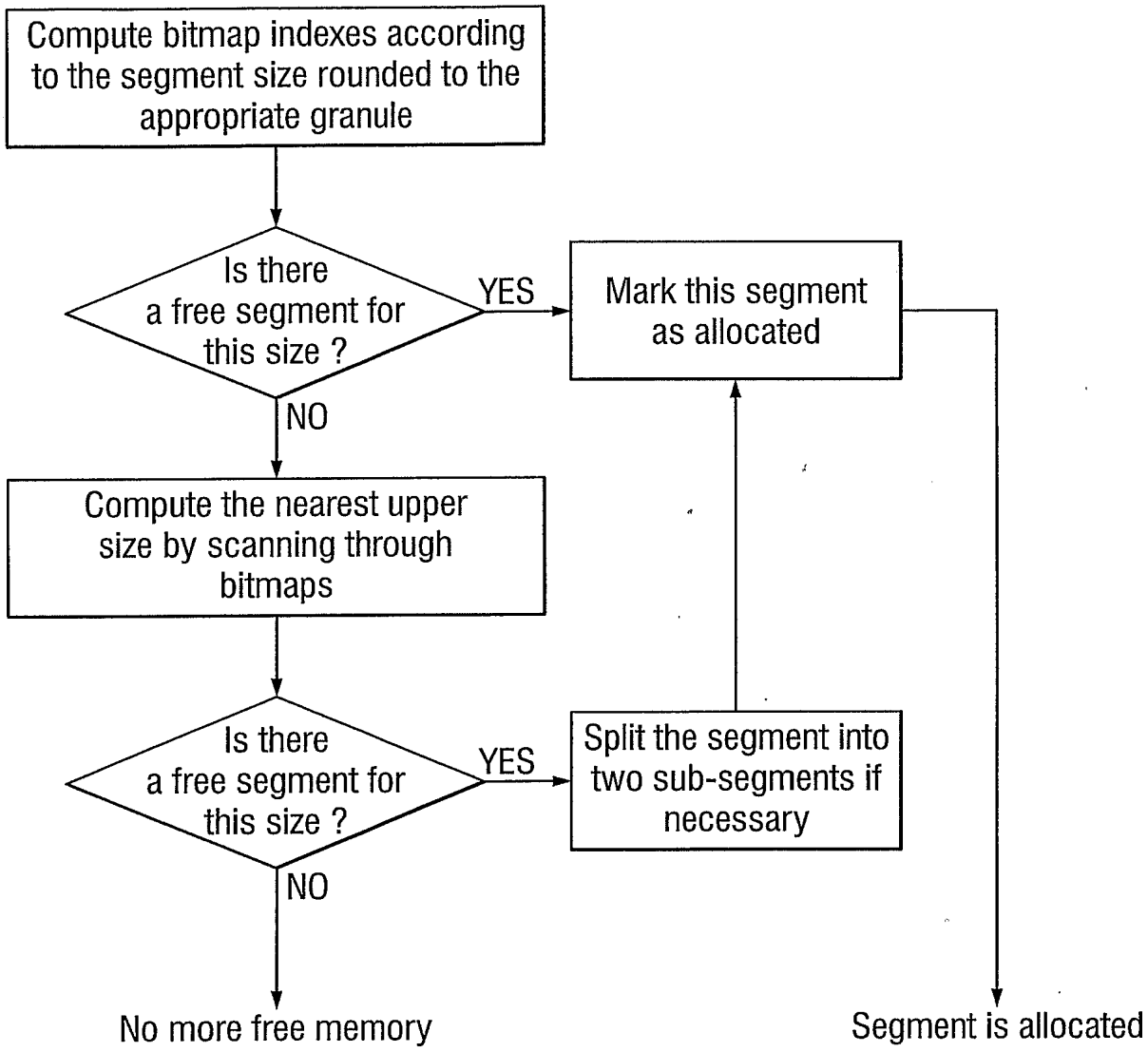
Compute bitmap indexes according to the segment size rounded to the appropriate granule
Is there a free segment for this size ?
YES
Mark this segment as allocated
NO
Compute the nearest upper size by scanning through bitmaps
Is there a free segment for this size ?
YES
Split the segment into two sub-segments if necessary
NO
No more free memory
Segment is allocated

Fig: 3

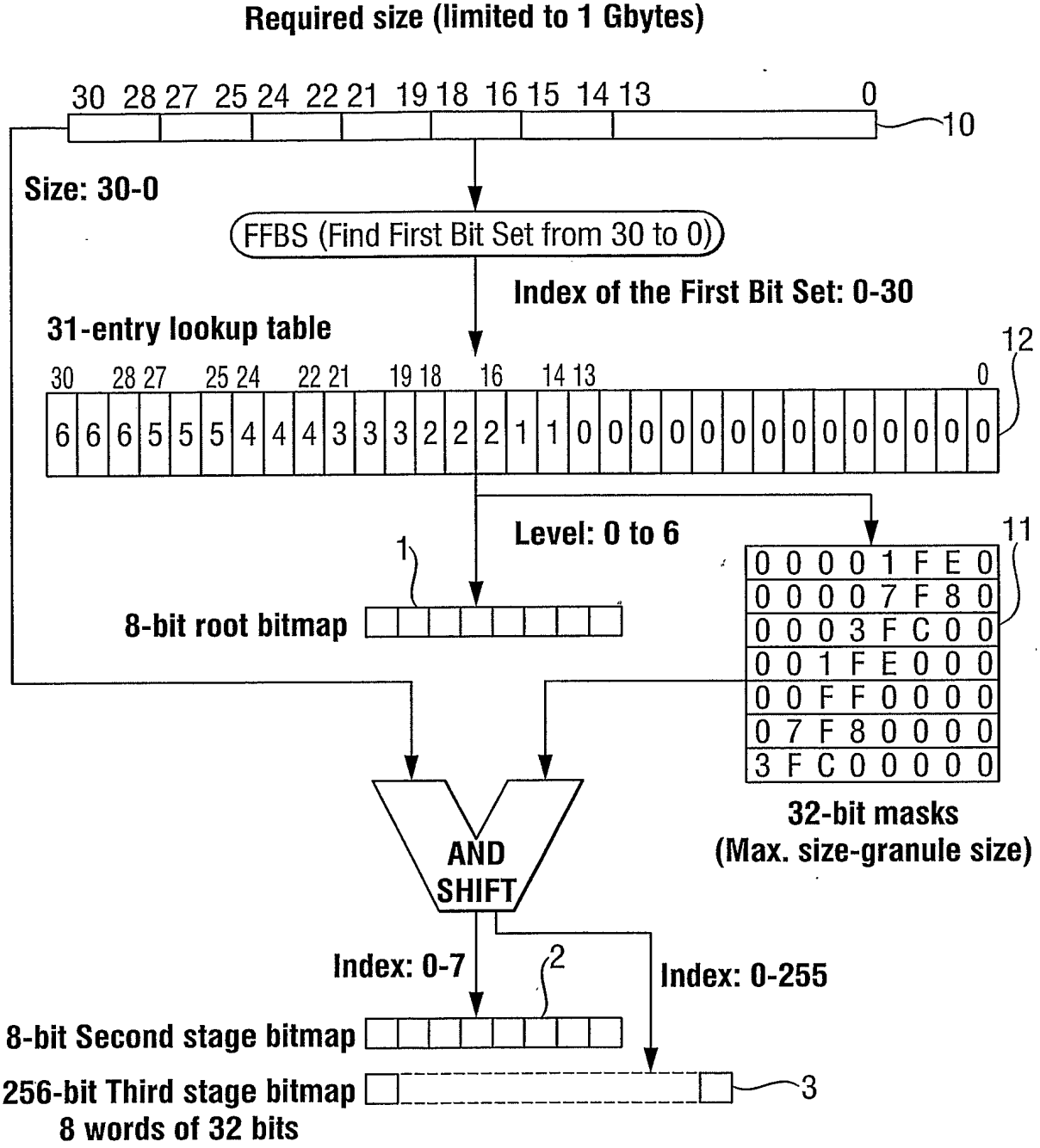
Required size (limited to 1 Gbytes)
30 28 27 25 24 22 21 19 18 16 15 14 13 0
10
Size: 30-0
FFBS (Find First Bit Set from 30 to 0)
Index of the First Bit Set: 0-30
31-entry lookup table
12
30 28 27 25 24 22 21 19 18 16 14 13 0
6 6 6 5 5 5 4 4 4 3 3 3 2 2 2 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0
Level: 0 to 6
11
1
8-bit root bitmap
0 0 0 0 1 F E 0
0 0 0 0 7 F 8 0
0 0 0 3 F C 0 0
0 0 1 F E 0 0 0
0 0 F F 0 0 0 0
0 7 F 8 0 0 0 0
3 F C 0 0 0 0 0
32-bit masks
(Max. size-granule size)
AND
SHIFT
Index: 0-7
2
Index: 0-255
8-bit Second stage bitmap
256-bit Third stage bitmap
3
8 words of 32 bits

Fig. 4

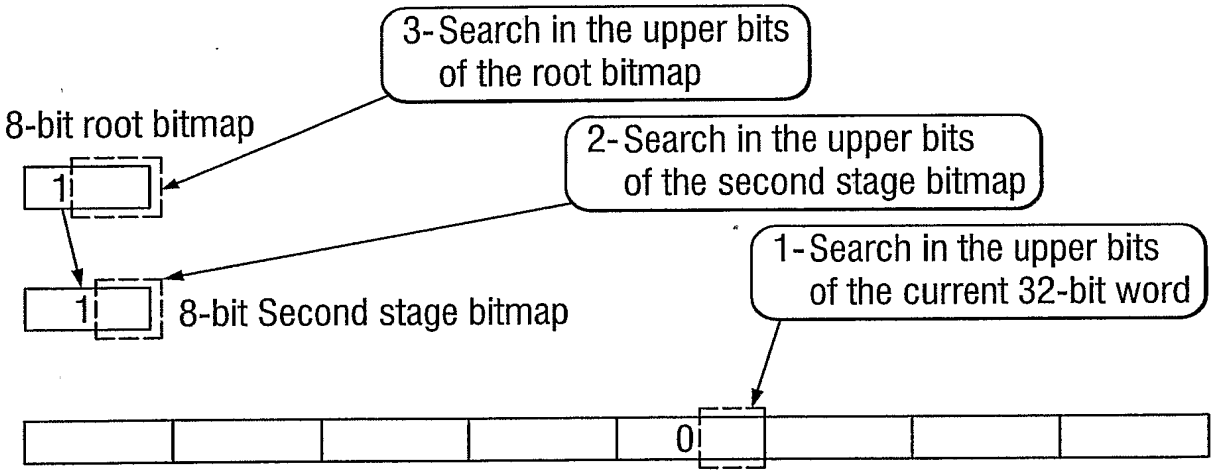
3-Search in the upper bits of the root bitmap
8-bit root bitmap
2-Search in the upper bits of the second stage bitmap
1-Search in the upper bits of the current 32-bit word
1
1
8-bit Second stage bitmap
0
256-bit Third stage bitmap organized as 8 words of 32 bits

Fig: 5

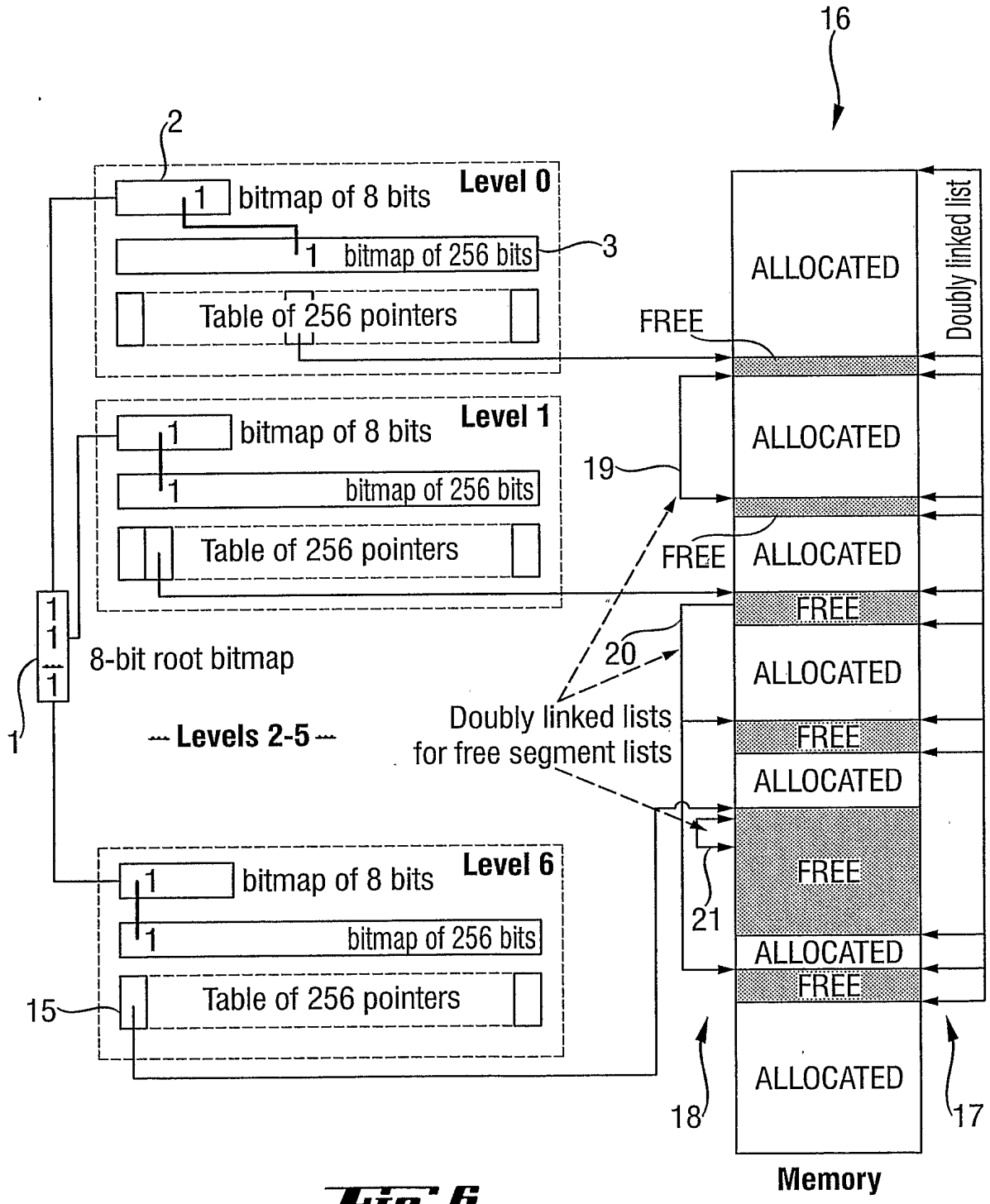
16
2
Level 0
1 bitmap of 8 bits
1 bitmap of 256 bits
3
Table of 256 pointers
Level 1
1 bitmap of 8 bits
1 bitmap of 256 bits
Table of 256 pointers
1
1
1
8-bit root bitmap
1
Levels 2-5
Level 6
1 bitmap of 8 bits
1 bitmap of 256 bits
15
Table of 256 pointers
Doubly linked lists
for free segment lists
19
20
21
ALLOCATED
FREE
Doubly linked list
ALLOCATED
FREE
ALLOCATED
FREE
ALLOCATED
FREE
ALLOCATED
FREE
ALLOCATED
FREE
ALLOCATED
18
17
Memory

Fig. 6

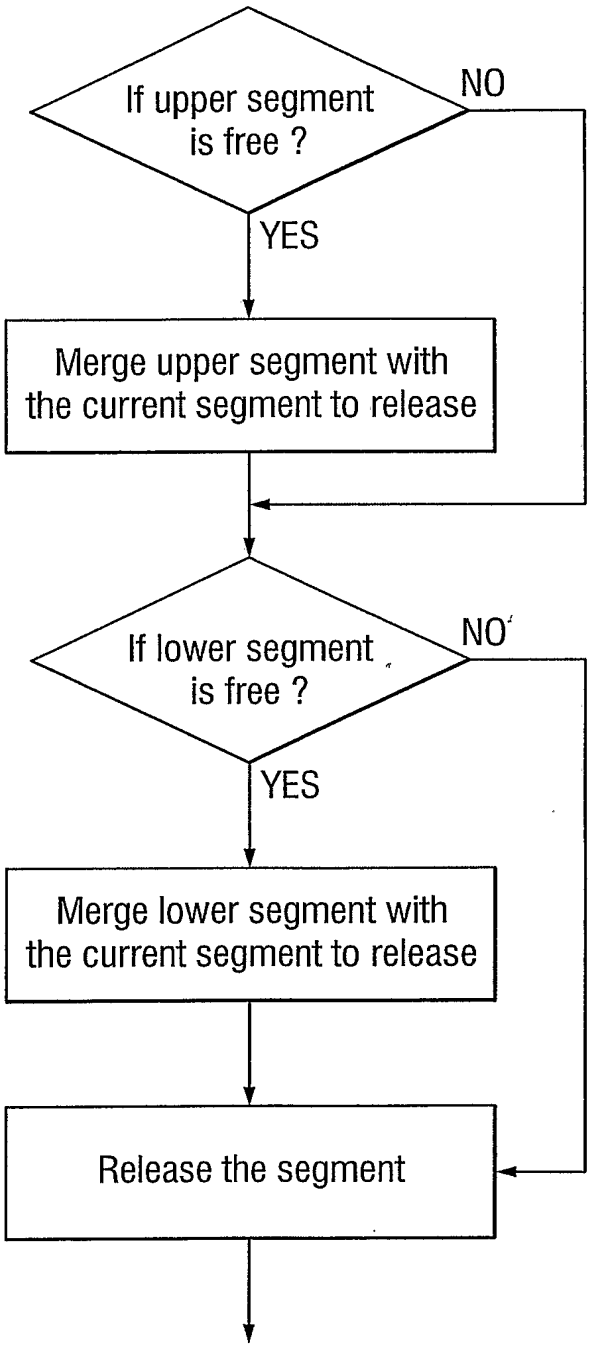
If upper segment is free ?
NO
YES
Merge upper segment with the current segment to release
If lower segment is free ?
NO
YES
Merge lower segment with the current segment to release
Release the segment

Fig: 1

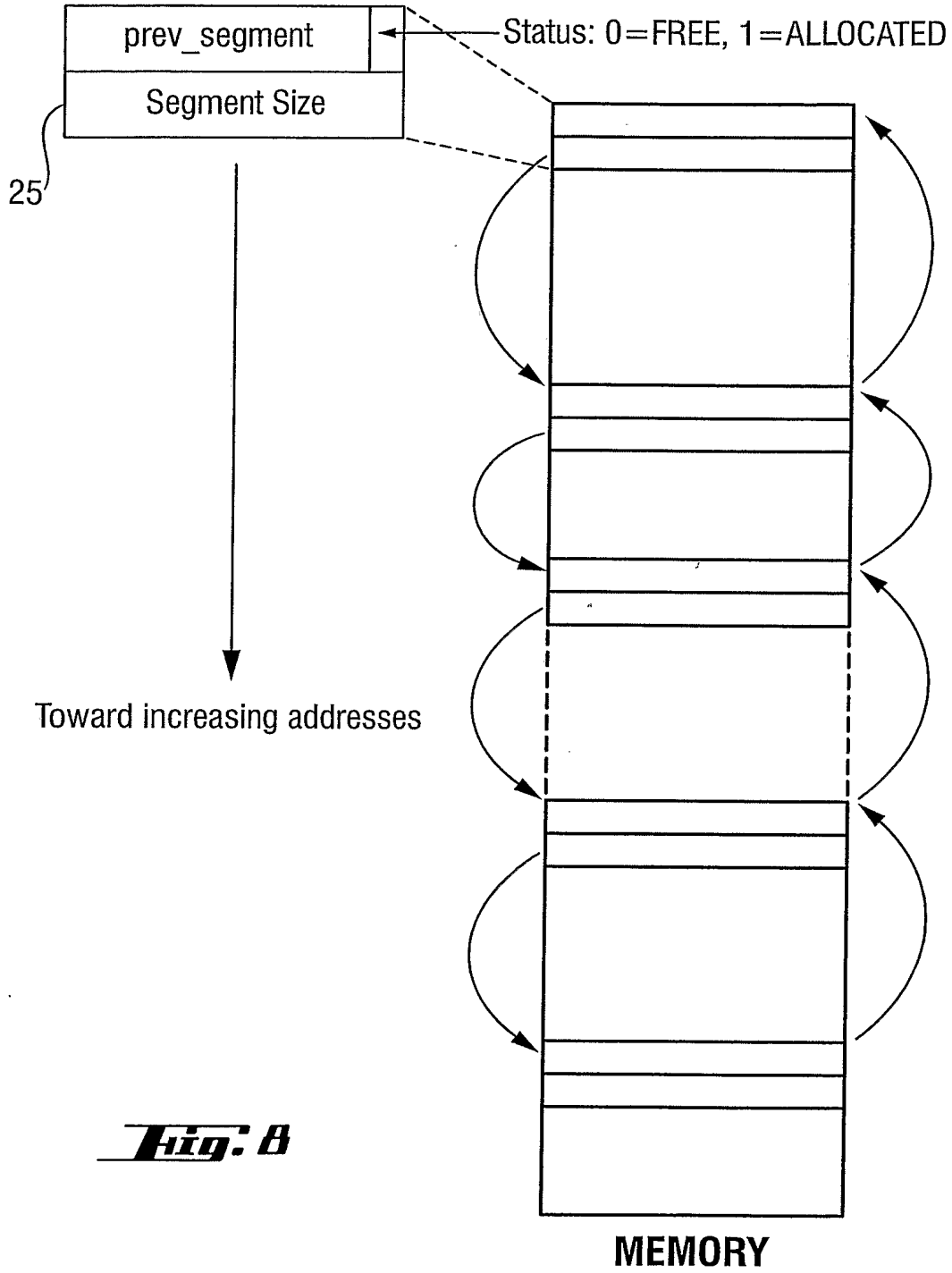
prev_segment
Segment Size
Status: 0=FREE, 1=ALLOCATED
25
Toward increasing addresses
MEMORY

Fig. 8

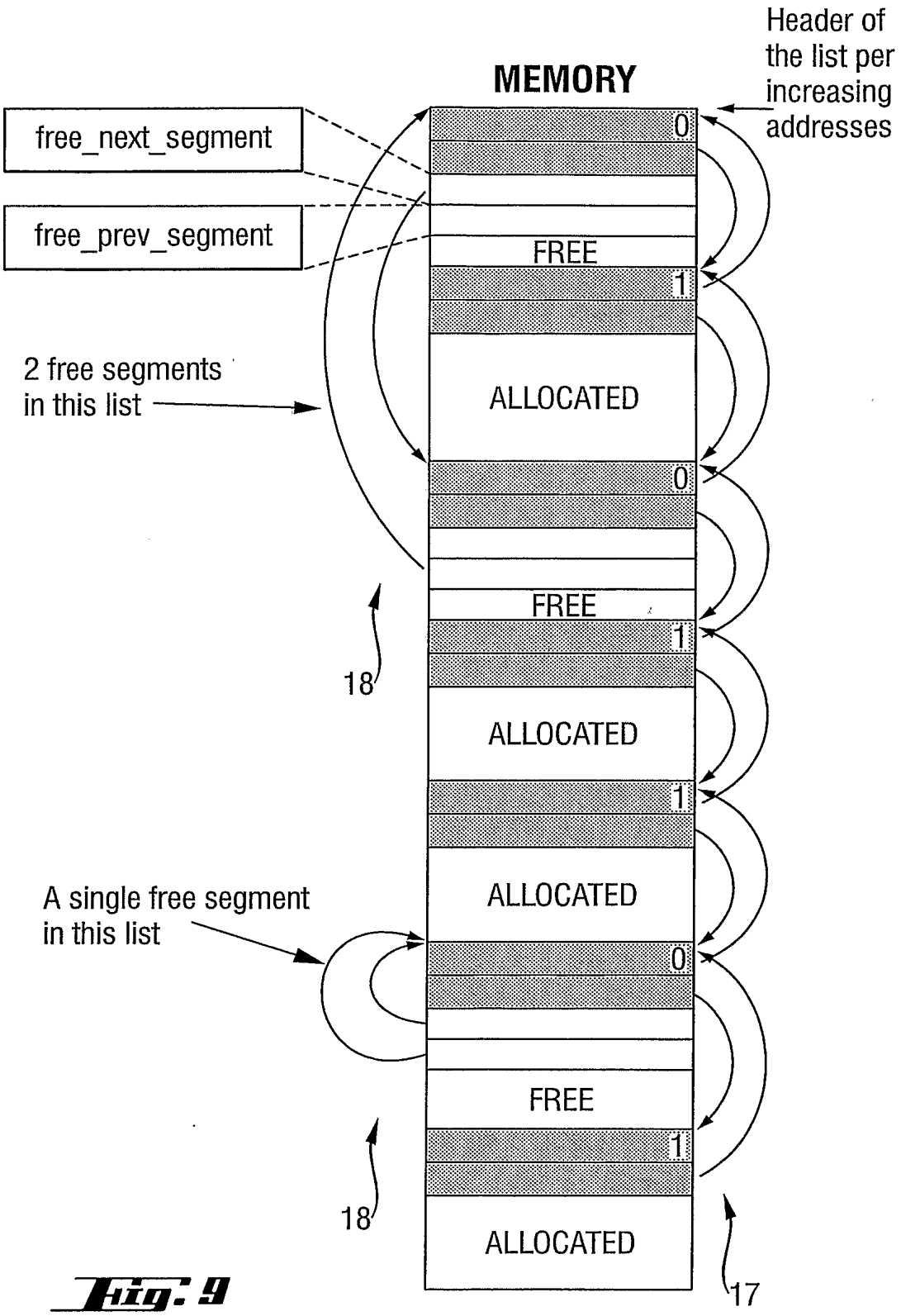
MEMORY
Header of the list per increasing addresses
free_next_segment
free_prev_segment
2 free segments in this list
A single free segment in this list
0
FREE
1
ALLOCATED
0
FREE
1
ALLOCATED
1
ALLOCATED
0
FREE
1
ALLOCATED
18
18
17

Fig. 9